```cpp
#include "Enemy.h"
#include <iostream>
#include <Windows.h>

Enemy::Enemy(int x, int y, int deltaX, int deltaY)
    : PlaceableActor(x, y, ActorColour::Green) // placing initial coordinates of enemy
    , m_currentMovementX(0)
    , m_currentMovementY(0)
    , m_directionX(0)
    , m_directionY(0)
    , m_movementInX(deltaX) //  The maximum distance the enemy can move in the x-direction
    , m_movementinY(deltaY) //  The maximum distance the enemy can move in the y-direction
{
    if (m_movementInX != 0)
    {
        m_directionX = 1;
    }
    if (m_movementinY != 0)
    {
        m_directionY = 1;
    }
}

void Enemy::Draw()
{
    HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(console, (int)m_colour);
    std::cout << (char)153; // prints coloured enemy.
    SetConsoleTextAttribute(console, (int)ActorColour::Regular);
}

void Enemy::Update() // update the state of the enemy
{
    if (m_movementInX != 0)
    {
        updateDirection(m_currentMovementX, m_directionX, m_movementInX);
    }
    if (m_movementinY != 0)
    {
        updateDirection(m_currentMovementY, m_directionY, m_movementinY);
    }

    this->SetXYPosition(m_pPosition->x + m_directionX, m_pPosition->y + m_directionY);
}

void Enemy::updateDirection(int& current, int& direction, int& movement) // responsible for handling the movement of the enemy
{
    current += direction;
```

```
49      if (std::abs(current) > movement) // reverse movement. if we reach the
          end we want to loop back the other way.
50      {
51          current = movement * direction;
52          direction *= -1; // change direction
53      }
54  }
55
56  //  If the absolute value of the current movement becomes greater than the
      maximum allowed movement (movement), it means the enemy has reached the end
       of its allowed movement range
```